



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/705,208	11/10/2003	Aaron Joseph McBride	2966-031366	2941

28289 7590 11/14/2008
THE WEBB LAW FIRM, P.C.
700 KOPPERS BUILDING
436 SEVENTH AVENUE
PITTSBURGH, PA 15219

EXAMINER

BUSS, BENJAMIN J

ART UNIT	PAPER NUMBER
----------	--------------

2129

MAIL DATE	DELIVERY MODE
-----------	---------------

11/14/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DETAILED ACTION

This Office Action is in response to an AMENDMENT entered 8/13/2008 for the patent application 10/705,208 filed on 11/10/2003. The Office Actions of 2/13/2008, 5/7/2007, 10/16/2006 and 4/20/2006 are fully incorporated into this Office Action by reference. Claims 1, 13-15, 22-35, & 40 are pending.

In the event that Applicant chooses to amend, the Examiner suggests clearly defining, in the claims, the following broad terms:

natural language	response layer	linked
input layer	recognized	tag
instruction	code	

Claim Objections***Response to Arguments***

Applicant's arguments, see page 7, filed 8/13/2008, with respect to the claim objection have been fully considered and are persuasive. The objection to claims 1 and 35 has been withdrawn.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1, 13-15, 23-24, 27-35, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chikirivao** (USPAP 2003/0163783) and **Wallace** ("The Elements of AIML Style").

Art Unit: 2129

Independent Claim 1:**Chikirivao** teaches:

- providing a template interface to the administrator, wherein the template includes at least one field to elicit information from the administrator (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository” ¶¶38-39),
- receiving information from the administrator into the template (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository” ¶¶38-39 and “entering and saving of data into a template” ¶43), and
- making the information accessible to a rules-based program for use in providing the at least one response in reply to a request from a user (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29), wherein the step of making the information accessible to the rules-based program saves the information as part of the template into rules (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository” ¶¶38-39 and “entering and saving of data into a template” ¶43 and “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29 and “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29; *The rule repository is clearly structured data storage*), and wherein the step of saving the information into rules includes the steps of:
 - o retrieving rules (pages 1-7 especially “system obtains the rule” ¶¶38-39),
 - o for each rule retrieved, determining whether the rule needs information (pages 1-7 especially “administrator may need to specify more or less information” ¶40 and “rules which are generated based upon ... information ... based upon parameters specified” ¶31 and “information and/or sub-rules needed to make such determinations” ¶32 and “rules may be designed with any level of interactivity and/or user

knowledge required and may include and utilize data and other information" ¶33 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48),

- if the rule needs information, retrieving the information from a corresponding field in the template and inserting the information into the rule (pages 1-7 especially "extracts from the provided information those parameters required by the rule(s)" ¶44-48 and "administrator may need to specify more or less information" ¶40 and "routing of information based upon the input template" ¶59),
- wherein the step of determining whether the rule needs information includes determining whether either a response layer or a logic layer needs information (pages 1-7 especially "rule which requires the user to provide inputs as to specific needs" ¶35 and "administrator may need to specify more or less information" ¶40 and "rules which are generated based upon ... information ... based upon parameters specified" ¶31 and "information and/or sub-rules needed to make such determinations" ¶32 and "rules may be designed with any level of interactivity and/or user knowledge required and may include and utilize data and other information" ¶33 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48), and

- retrieving information indicated as needed from a corresponding field in the template and inserting the information into the response layer or logic layer, as called for by the signifier (pages 1-7 especially "templates and other features that enable a user to expeditiously enter the necessary information required" ¶43 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48 and "information is received from the field" ¶50 and "routing of information based upon the input template" ¶59 and "administrator may need to specify more or less information" ¶40);

Chikirivao fails to teach:

- wherein the step of determining whether a layer needs information, includes the step of identifying a signifier in the layer,
 - wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select in the information from a corresponding field in the template so that the information will be linked to the rule, and
 - wherein the logic layer is configured to choose between various responses provided by the administrator,

Art Unit: 2129

- wherein at least one of the responses is recognized by the logic layer,
- wherein the chosen response is the response to be used in the response layer.

Wallace teaches:

- providing a template interface to the administrator, wherein the template includes at least one field to elicit information from the administrator (p1-83 especially i.e. "template window" p42; *Examiner acknowledges that the term "<template>" in the reference is closer to the claimed "rules" than the claimed "template". The "template window" on page 42 of the reference does read on the claimed "template" as it is "interface provided to the administrator / botmaster in order to elicit information that will become part of the rules from which a rules-based program provides responses*),
- receiving information from the administrator into the template (p1-83 especially i.e. "In the template window, the botmaster types: ..." p42-43 or "Now s/he edits the template to read: ..." p44), and
- making the information accessible to a rules-based program for use in providing the at least one response in reply to a request from a user, wherein the step of making the information accessible to the rules-based program saves the information as part of the template into rules (p1-83 especially "store" p30 or "save" p43-44 or p47 or p50 or p53 p79), and wherein the step of saving the information into rules includes the steps of:
 - retrieving rules (p1-83 especially i.e. "categories" p12-13 or "patterns and templates" p21-23),
 - for each rule retrieved, determining whether the rule needs information (p1-83 especially "AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories" p12-13; *Examiner points out that AIML tags indicate when the rules need more information*),
 - if the rule needs information, retrieving the information from a corresponding field in the template and inserting the information into the rule (p1-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57),
 - wherein the step of determining whether the rule needs information includes determining if either a response layer or a logic layer needs information by identifying the presence of a signifier in the response layer or the logic layer, respectively, wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select the information from a corresponding field in the template so that the information will be linked to the rule (p1-83 especially "AIML tags transform the reply into a mini computer program which can save data,

activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories" p12-13 or p53-55; *Examiner points out that AIML tags indicate when the rules need more information. These tags can be part of the logic layer used to determine the appropriate branches to take in seeking a response for the input, or they can be part of the response layer used to give responses or call processes in response to the input*), and

- wherein the logic layer is configured to choose between various responses provided by the administrator (p1-83 especially i.e. p53-55 or p56-57),
- wherein at least one of the responses is recognized by the logic layer (p1-83 especially i.e. p21-23 or p52-53; *Examiner points out that AIML relies on the graph master / logic layer recognizing which administrator provided response is appropriate for the given input*),
- wherein the chosen response is the response to be used in the response layer (p1-83 especially i.e. "The algorithm finds best-matching pattern for each input. The category ties the response template directly to the stimulus pattern" p38-39 or p56-57), and
- retrieving information indicated as needed from a corresponding field in the template and inserting the information into the response layer or logic layer, respectively (p1-83 especially i.e. p12-13 or p16-17 or p56-57);

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by identifying a signifier / tag in a layer such as the response or logic layer to determine whether information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Independent Claim 35:

Chikirivao teaches:

- An interface configured to receive information from the administrator (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "querying the administrator ... create a customized rule based upon a pre-existing

Art Unit: 2129

customizable rule template saved in the rule repository" ¶38-39 and "entering and saving of data into a template" ¶43);

- A template accessible to the administrator, wherein the template includes at least one field to elicit information from the administrator (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "entering and saving of data into a template" ¶43);
- An engine configured to:
 - o Make the information accessible to a rules-based program that provides the at least one response in reply to the inputs from the user (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" ¶38-39 and "rules created by an administrator are preferably saved in the rule repository" ¶29);
 - o Retrieve the rules (pages 1-7 especially "system obtains the rule" ¶38-39);
 - o For each rule retrieved, determine whether the rule needs information (pages 1-7 especially "administrator may need to specify more or less information" ¶40 and "rules which are generated based upon ... information ... based upon parameters specified" ¶31 and "information and/or sub-rules needed to make such determinations" ¶32 and "rules may be designed with any level of interactivity and/or user knowledge required and may include and utilize data and other information" ¶33 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48);
 - o Retrieve the information from a corresponding field in the template and insert the information into the rule if the rule needs information (pages 1-7 especially "extracts from the provided information those parameters required by the rule(s)" ¶44-48 and "administrator may need to specify more or less information" ¶40 and "routing of information based upon the input template" ¶59);
 - o Determine if either a response layer or a logic layer needs information (pages 1-7 especially "rule which requires the user to provide inputs as to specific needs" ¶35 and "administrator may need to specify more or less information" ¶40 and "rules which are generated based upon ... information ... based upon parameters specified" ¶31 and "information and/or sub-rules needed to make such determinations" ¶32

and "rules may be designed with any level of interactivity and/or user knowledge required and may include and utilize data and other information" ¶33 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48); and

- retrieving information indicated as needed from a corresponding field in the template and inserting the information into the response layer or logic layer, respectively (pages 1-7 especially "templates and other features that enable a user to expeditiously enter the necessary information required" ¶43 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48 and "information is received from the field" ¶50 and "routing of information based upon the input template" ¶59 and "administrator may need to specify more or less information" ¶40);

Chikirivao fails to teach:

- Wherein the step of determining whether a layer needs information, includes identifying a signifier in the layer,
 - o wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select in the information from a corresponding field in the template so that the information will be linked to the rule, and
 - o wherein the logic layer is configured to choose between various responses provided by the user,
 - o wherein at least one of the responses is recognized by the logic layer,
 - o wherein the chosen response is the response to be used in the response layer.

Wallace teaches:

- an interface configured to receive information from the administrator (p1-83 especially i.e. "In the template window, the botmaster types: ..." p42-43 or "Now s/he edits the template to read: ..." p44);
- a template accessible to the administrator, wherein the template includes at least one field to elicit information from the administrator (p1-83 especially i.e. "In the template window, the botmaster types: ..." p42-43 or "Now s/he edits the template to read: ..." p44; *Examiner acknowledges that the term "<template>" in the reference is closer to the claimed "rules" than the claimed "template". The "template window" on page 42 of the reference does read on the claimed "template" as it is "interface provided to the administrator / botmaster in order to elicit information that will become part of the rules from which a rules-based program provides responses); and*
- an engine configured to:

Art Unit: 2129

- make the information accessible to a rules-based program that provides the at least one response in reply to the inputs from the user (p1-83 especially p43-44 or p47 or p53);
- save the information as part of the template into rules (p1-83 especially i.e. "store" p30 or "save" p43-44 or p47 or p50 or p53 p79);
- 5 ○ retrieve the rules (p1-83 especially i.e. "categories" p12-13 or "patterns and templates" p21-23);
- for each rule retrieved, determine whether the rule needs information (p1-83 especially i.e. "AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories" p12-13; *Examiner points out that AIML tags indicate when the rules need more information*);
- 10 ○ retrieve the information from a corresponding field in the template and insert the information into the rule if the rule needs information (p1-83 especially i.e. p12-13 or p21-23 or p38-39 or p56-57);
- determine if either a response layer or a logic layer needs information by identifying the presence of a signifier in the response layer or the logic layer, respectively, wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select the
- 15 information from a corresponding field in the template so that the information will be linked to the rule (p1-83 especially i.e. "AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories" p12-13 or p53-55; *Examiner points out that AIML tags indicate when the rules need more information. These tags can be part of the logic layer used to determine the*
- 20 *appropriate branches to take in seeking a response for the input, or they can be part of the response layer used to give responses or call processes in response to the input*), and
- wherein the logic layer is configured to choose between various responses provided by the administrator (p1-83 especially i.e. p53-55 or p56-57),
- wherein at least one of the responses is recognized by the logic layer (p1-83 especially i.e. p21-23 or
- 25 p52-53; *Examiner points out that AIML relies on the graph master / logic layer recognizing which administrator provided response is appropriate for the given input*),

Art Unit: 2129

- wherein the chosen response is the response to be used in the response layer (p1-83 especially i.e. "The algorithm finds best-matching pattern for each input. The category ties the response template directly to the stimulus pattern" p38-39 or p56-57), and
- retrieve information indicated as needed from a corresponding field in the template and inserting the information into the response layer or logic layer, respectively (p1-83 especially i.e. p12-13 or p16-17 or p56-57);

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by identifying a signifier / tag in a layer such as the response or logic layer to determine whether information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Claim 13:

Chikirivao fails to teach:

- Wherein the signifier is a tag in a text string.

Wallace teaches:

- Wherein the signifier is a tag in a text string (p1-83 especially i.e. "AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories" p12-13 or p15 or p40 or p53-55).

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by identifying a signifier / tag in a text string to determine whether information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Art Unit: 2129

Claim 14:

Chikirivao fails to teach:

- Wherein the signifier is an instruction embedded in a text string.

Wallace teaches:

- 5 - Wherein the signifier is an instruction embedded in a text string (p1-83 especially i.e. p12-13 or p15-18).

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by identifying a instruction embedded in a text string to determine whether information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Claim 15:

Chikirivao fails to teach:

- 15 - Wherein the signifier is a code.

Wallace teaches:

- Wherein the signifier is a code (p1-83 especially i.e. "<system> tag executes any program accessible as an operating system shell command, and inserts the results in the reply. Similarly, the <javascript> tag allows arbitrary scripting" p12-13 or p15-18).

20 Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by identifying a code to determine whether information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Art Unit: 2129

Claim 23:

Chikirivao teaches:

- Wherein the step of making the information accessible to the rules-based program is accomplished by receiving a manual command from a user (pages 1-7 especially “access to a rule may be specified manually or automatically” ¶43).

Claim 24:

Chikirivao teaches:

- Wherein the step of making the information accessible to the rules-based program is accomplished automatically upon the occurrence of a predefined event (pages 1-7 especially “access to a rule may be specified manually or automatically” ¶43).

Claim 27:

Chikirivao teaches:

- Wherein the predefined event is activation of a save function by the administrator (pages 1-7 especially “access to a rule may be specified manually or automatically ... rule may be activated upon the entering and saving of data into a template” ¶43).

Claims 28 and 40:

Chikirivao teaches:

- Further including the step of enabling the administrator to edit the information (pages 1-7 especially “querying the administrator ... modify an existing rule ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... allow the user to modify/customize the rule” ¶38-39 and “enables such administrators to ... edit ... the rules” ¶29).

Claim 29:

Chikirivao teaches:

- Wherein the step of enabling the administrator to edit the information includes the steps of:

Art Unit: 2129

- Retrieving the information (pages 1-7 especially “obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule” ¶¶38-39),
- Posting the information in at least one appropriate field in the template (pages 1-7 especially “based upon a pre-existing customizable rule template ... obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule” ¶¶38-39 and “templates and other features that enable a user to expeditiously enter the necessary information required for a given task” ¶43),
- Receiving edited information from the administrator into the template (pages 1-7 especially “based upon a pre-existing customizable rule template ... obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule” ¶¶38-39 and “templates and other features that enable a user to expeditiously enter the necessary information required for a given task” ¶43), and
- Making the edited information accessible to the rules-based program for use in providing the at least one response in reply to a request from the user (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29).

Claim 30:**Chikirivao** teaches wherein:

- The step of making the information accessible to the rules-based program saves the information as part of the template (pages 1-7 especially “access to a rule may be specified manually or automatically ... rule may be activated upon the entering and saving of data into a template” ¶43), and
- The step of retrieving the information includes the step of restoring the information to the at least one field (pages 1-7 especially “based upon a pre-existing customizable rule template ... obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule” ¶¶38-39).

Art Unit: 2129

Claim 31:**Chikirivao** teaches wherein:

- The step of making the information accessible to the rules-based program saves the information as structured data (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29; *The rule repository is clearly structured data storage*).

Chikirivao fails to teach wherein:

- The step of retrieving the information includes the steps of, for at least one of the at least one field in the template:
 - o Retrieving instructions indicating where the information is stored, and
 - o Executing the instructions to retrieve the information.

Wallace teaches wherein:

- The step of making the information accessible to the rules-based program saves the information as structured data (p1-83 especially i.e. “structure” p21-24 or p30 or p50 or p79), and
- The step of retrieving the information includes the steps of, for at least one of the at least one field in the template:
 - o Retrieving instructions indicating where the information is stored (p1-83 especially i.e. p21-24)
 - o Executing the instructions to retrieve the information (p1-83 especially i.e. p12-13 or p21-24 or p38-39 or p56-57).

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by using executing instructions to retrieve information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Art Unit: 2129

Claim 32:**Chikirivao** teaches wherein:

- The step of making the information accessible to the rules-based program saves the information into rules (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29).

Chikirivao fails to teach wherein:

- The step of retrieving the information includes the steps of, for at least one of the at least one field in the template:
 - o Retrieving instructions indicating where the information is stored, and
 - o Executing the instructions to retrieve the information.

Wallace teaches wherein:

- The step of making the information accessible to the rules-based program saves the information as structured data (p1-83 especially i.e. “structure” p21-24 or p30 or p50 or p79), and
- The step of retrieving the information includes the steps of, for at least one of the at least one field in the template:
 - o Retrieving instructions indicating where the information is stored (p1-83 especially i.e. p21-24)
 - o Executing the instructions to retrieve the information (p1-83 especially i.e. p12-13 or p21-24 or p38-39 or p56-57).

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by using executing instructions to retrieve information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Claim 33:**Chikirivao** teaches wherein:

- The step of making the information accessible to the rules-based program saves the information into rules (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable

Art Unit: 2129

rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29).

Chikirivao fails to teach wherein:

- The step of retrieving the information includes the steps of, for each rule used:
 - 5 o Determining whether the rule includes a signifier, and
 - o If a signifier is included, executing instructions from the signifier to retrieve the information associated with the rule.

Wallace teaches wherein:

- The step of retrieving the information includes the steps of, for each rule used:
 - 10 o Determining whether the rule includes a signifier (-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57), and
 - o If a signifier is included, executing instructions from the signifier to retrieve the information associated with the rule (-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57).

Rationale:

15 **Chikirivao** and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by using executing instructions to retrieve information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Claim 34:

Chikirivao teaches wherein:

- The step of making the information accessible to the rules-based program saves the information into rules (pages 1-7 especially “querying the administrator ... create a customized rule based upon a pre-existing customizable
- 25 rule template saved in the rule repository ... either testing the rule or saving the rule” ¶¶38-39 and “rules created by an administrator are preferably saved in the rule repository” ¶29).

Chikirivao fails to teach wherein:

- The step of retrieving the information includes the steps of, for each rule used:

Art Unit: 2129

- Determining whether the rule includes a signifier, and
- If a signifier is included, retrieving the information tagged in the rule.

Wallace teaches wherein:

- The step of retrieving the information includes the steps of, for each rule used:

- Determining whether the rule includes a signifier (-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57), and
- If a signifier is included, executing instructions from the signifier to retrieve the information associated with the rule (-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57).

Rationale:

Chikirivao and **Wallace** are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of **Chikirivao** by using executing instructions to retrieve information is needed as taught by **Wallace** for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (**Wallace** p36, 77-80).

Response to Arguments

Applicant's arguments filed 8/13/2008 have been fully considered but they are not persuasive. In re page 7-8, applicant argues:

The Wallace publication fails to disclose a "template" and/or a "logic layer" consistent with the claims and across both the Chikirivao publication and the Wallace publication. The Examiner acknowledges on page 5 of the Office Action how "the term "<template>" in the Wallace publication is closer to the claimed "rules" than the claimed "template"." The claims have limitations encompassing the terms "rules" and "templates", thereby indicating that the terms are literally distinguished, as they serve different roles and embody different concepts. Thus, it can be seen that an inconsistent reading of the disclosure of the Wallace publication is applied to the claims at issue.

Specifically, in §5(a) of his Declaration, Mr. Keane declares:

[A]s set forth in the present application, the purpose of a template, is a way to create a specific rule or rules, based on a pre-defined form (the template), containing markers for additional information needed to define the rule (signifiers), as provided by the administrator. In the Wallace publication, AIML tags, which are equated by the Examiner with the Applicants' signifiers, serve a very different role, in that they are only activated during the execution of the AIML rules to control the flow of the program defined by those rules. AIML tags are an exclusively run-time control structure. In contrast, the signifiers in the Applicants' templates, are an exclusively compile-time structure, as they are used to construct run-time rules. There is no disclosure in the Wallace publication with respect to a mechanism

Art Unit: 2129

that would correspond to a compilation of run-time roles from a partially-defined template. Every feature described in the Wallace publication is part of the run-time system. The use of the term "template", which appears in AIML, means something completely different, as it is the term used to describe the form of an output or reply in a rule. Therefore, it is not the case that AIML tags denote places where rules "need information", as an AIML tag denotes a control branch in the execution of the rule, which may cause recursion, output, or even external code execution.

Examiner disagrees. The claimed "template" and "logic layer" are both taught as detailed in the rejection above. The examiner has not indicated that the term "<template>" in **Wallace** corresponds to the claimed template. The examiner has established that the "template window" (**Wallace**, i.e. p42) does act as the claimed "template interface" in which a "template includes at least one field to elicit information from the administrator". The "template window" of **Wallace** is an interface provided to the administrator/botmaster in order to elicit information that will become part of the rules from which a rules-based program provides responses. In other words, the "template window" of **Wallace** is used as a way to create specific rules based on a predefined form, as required by the claims.

The claimed "logic layer" is required to be able to recognize a response and to choose between various responses. Examiner asserts that any branching program logic would meet this requirement. For example, in **Wallace**, a response is recognized as a potential response based on the tags that enclose it -- it is determined to be a valid potential response based on the category and pattern with which it is associated.

The claims do not require any features to be "compile-time" features. It is well known in the art to have computer languages that are scripted or interpreted rather than compiled, such as ASP, BASIC, COBOL, HTML, XML, MATLAB, PHP, Ruby, Smalltalk, etc.

Applicant's arguments are narrower than the claims require. In re pages 8-9, applicant argues:

The Examiner also equates AIML tags with the "signifiers" set forth in the claims. However, equating the two fails to take into account the substantial differences in purpose and action between signifiers and AIML tags. Contrary to the Examiner's assertion, AIML tags do not indicate that a rule "needs information." Instead, AIML tags simply act as programming directives, as supported by the disclosure of the Wallace publication on page 12 reprinted below:

More generally, AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories.

Most AIML tags in fact belong to this template side sub language.

The Wallace publication refers to tags as a "sub language". AIML tags do not have the property that causes the information to be "linked to the rule", as is required by the claims. The signifiers of the present invention are markers that identify a specific piece of information "such that the call for information invokes a process

Art Unit: 2129

to select the information from a corresponding field in the template so that the information will be linked to the rule."

5 Examiner disagrees. It is true that the examiner has equated AIML tags with the claimed "signifier". Examiner maintains that these AIML tags do indicate that information is needed from elsewhere. Though they are programming directives or a "sub language", they meet the requirements of the claim limitations. Linking information to the rule, in the broadest reasonable sense, is establishing any connection between the rule and the information. The AIML tags do this by indicating it is appropriate to have a conditional response, indicating a recursive call needs to be made, etc. The AIML tags may indicate information is needed from another function (<srai>) or by picking from a finite set of choices (<random>). That <random> indicates choosing randomly does not indicate any lack of logic, so long as the random selection is between equally valid responses. Indeed, in that situation, a random selection is the most logical choice, given that there is no logical difference between the responses. Examiner maintains that the AIML tags indicate that more information is required – that something needs to be done to pull in information. In re page 9, applicant argues:

15 With respect to the logic layer limitation of the claims, Mr. Keane declares in §5(b) of his Declaration:

20 The logic layer of the present invention implements a distinct processing step whereby a specific response can be selected from a set of equivalent responses in a principled manner, which may reference information not present in the user input to the system. Neither the Chikirivao nor Wallace publications show this feature. The Wallace publication describes AIML, however, AIML is a language without the explicit notion of the claimed logic layer, as responses in AIML proceed directly from rule input matching, or recursion on the input matching. It is incorrect to equate the "Graphmaster" of the Wallace publication to the claimed logic layer because the Graphmaster is a representation of the input space matching capabilities of a particular AIML definition set (e.g., A.L.I.C.E.). There is no explicit logic layer in AIML, as the output is determined by the user input and the rules. In the present invention, the matching of an input is only the first step to determining the appropriate response, followed by the processing of the logic layer.

35 The AIML model does not have an explicit logic layer, as found in the present invention. The <random> tag grouping in AIML allows an implicit variation among equivalent outputs through random choice. There is no disclosure, teaching, or suggestion in the Wallace publication of a mechanism equivalent to the claimed logic layer that is able to take additional information, if needed, and perform a defined computation that can determine the selection of an output from among a set of outputs.

40 In AIML, the <srai> tag allows for a recursive rule definition (See page 13 of the Wallace publication). While this allows considerable flexibility in the form of rule definition, it does not, in and of itself, introduce any functionality in the matching power of AIML that could not be replicated by finite-length non-recursive matching rules. The <srai> tag does not provide the concept of choice over several possible outcomes all appropriate for a particular input, but

45

Art Unit: 2129

selectable by additional information not contained in the input.

Examiner disagrees. The broadest reasonable interpretation of the claim language certainly doesn't require that the logic layer *understand* the possible responses. **Wallace** discusses how the AIML interpreter is capable of logically deducing appropriate responses based on patterns and associations (**Wallace**, i.e. p55) – thereby filling in needed information through inference as guided by the tags. Applicant's arguments neglect to consider that every quest for an answer begins with every piece of knowledge in the system, which AIML refines based on a tracked category, then by pattern matching. These are all logical operations which select an answer among many possibilities. That a final selection may be made randomly is no less logical so long as the possible answers from which the selection is made are equally valid. That output is determined by subjecting user inputs to a set of rules is evidence of a logical processing of the input by a layer of AIML -- thus the claimed "logic layer" is met. In re page 10, applicant argues:

Accordingly, the claimed concept of a "template" and "logic layer" in the context of a consistent reading of the other limitations of the independent claims and the claims depending therefrom is not disclosed, taught, or suggested in any of the prior art of record. In light of the aforementioned arguments made with respect to the anticipation rejections under the Wallace publication, whose underlying anticipation teachings, now refuted, are used for rejecting at least the independent claims on an obviousness basis in view of the teaching of the Chikirivao publication, Applicants hereby respectfully requests that the Examiner withdraw the overall obviousness rejections.

Examiner disagrees. This Office Action clearly establishes, both in the arguments above and in the detailed rejection above, that the claimed "template", "logic layer", and "signifier" are met by the prior art of record. In re page 10, applicant argues:

When undertaking an obviousness analysis, the Examiner is also required to take into account secondary considerations relation to applicant's invention. *Graham v. John Deere*, 383 U.S. 1, 148 USPQ 459 (1966). The Court of Appeals for the Federal Circuit stated in *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 1538, 218 USPQ 871,879 (Fed. Cir. 1983) that "evidence rising out of the so-called 'secondary considerations', such as commercial success, must always when present be considered en route to a determination of obviousness" (See also *KSR v. Teleflex*, 550 U.S. ___ at 2). In {}3 of his Declaration, Mr. Keane declares:

The claimed invention was first commercialized in July of 2003. The Assignee has commercially pursued deployments that incorporate the claimed invention. With respect to enterprise deployments, the Assignee [has] offered, since as early as 2005 and through the present, implementations containing the claimed invention to various companies including PSEG (a major Northeastern U.S. electric and gas utility company), Qantas Airlines (the leading Australian air carrier), and the University of Phoenix (one of the largest higher educational institutions in the world, enrolling approximately 400,000 students). Currently, the Assignee is in contract negotiations to deploy its product through Sutherland Global Services, a company providing business process outsourcing services to Fortune 500 companies, including Dell. The estimated total sales of products and deployments that incorporate the

Art Unit: 2129

claimed invention are at least \$150,000. The foregoing information is indicative of the commercial success of the claimed invention.

As set forth in §4 of the Declaration, customers utilizing the claimed invention reduce their costs while improving their online customer service. As discussed in the Declaration, as an example, cumbersome manual processes have been replaced with automation-assisted online processes, thereby reducing errors and increasing customer satisfaction.

Examiner disagrees. Examiner has considered the evidence rising out of the so-called "secondary considerations", but has not found it to be persuasive. Applicant has not proven that the commercial success is result of a direct correspondence with the claimed "template" and/or "logic layer" and/or "signifier." As the commercial success is not explicitly tied the disputed limitations, the examiner is not persuaded that any of these features are directly responsible for the commercial success. Therefore, the commercial success has been considered, but is not persuasive. In re page 11, applicant argues:

For the foregoing reasons, Applicants believes that the subject matter of independent claims 1 and 35 are not rendered obvious by the Chikirivao publication in view of the Wallace publication or any other prior art of record. Reconsideration of the rejections of independent claims 1 and 35 is respectfully requested. Claims 13-15 and 22-34 depend from and add further limitations to amended independent claim 1 and are believed to be patentable for the reasons discussed hereinabove in connection with amended independent claim 1. Claim 40 depends from and adds further limitations to amended independent claim 35 and is believed to be patentable for the reasons discussed hereinabove in connection with amended independent claim 35. Reconsideration of the rejections of dependent claims 13-15, 22-34, and 40 is respectfully requested.

Examiner disagrees. The rejection has been reviewed and applicant's arguments have been considered, but the examiner is not persuaded for the reasons detailed above. The outstanding rejection under 35 U.S.C. §103 is maintained.

Claim Rejections - 35 USC § 103

Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Chikirivao** (USPAP 2003/0163783) and **Wallace** ("The Elements of AIML Style") in view of **Jammes** (USPN 6,484,149).

Claim 22:

The combination of **Chikirivao** and **Wallace** fails to teach:

- Wherein the step of retrieving rules retrieves all of the rules in a template information script.

Jammes teaches:

Art Unit: 2129

- Wherein the step of retrieving rules retrieves all of the rules in a template information script (C1-56 especially “based on a template ... scripts to extract stored ... patterns ... against customization rules” C43:40-65).\

Motivation:

Jammes and the combination of **Chikirivao** and **Wallace** are from the same field of endeavor, software development. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the combined teachings of **Chikirivao** and **Wallace** by retrieving all of the rules in a template information script as taught by **Jammes** for the benefit of making the on-line experience more convenient and expedient as well as more pleasant (**Jammes** C4:10-35).

Response to Arguments

Applicant has not separately argued the claim in this rejection. The rejection of the claims upon which this claim depends has been maintained, and this rejection is maintained for the same reasons, as detailed above.

Claim Rejections - 35 USC § 103

Claims 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chikirivao** (USPAP 2003/0163783) and **Wallace** (“The Elements of AIML Style”) in view of **Habraken** (“Microsoft Office XP 8-in-1” – Part III: Word – Chapter 2: Working with Documents).

Claim 25:

The combination of **Chikirivao** and **Wallace** fails to teach:

- Wherein the predefined event is closing of the template.

Habraken teaches:

- Wherein the predefined event is closing of the template (pages 4-16 especially “Before closing ... asks whether you want to save these changes before closing” page 15).

Motivation:

Habraken and the combination of **Chikirivao** and **Wallace** are from the same field of endeavor, software. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the combined teachings of **Chikirivao** and **Wallace** by saving information to be available occurs when closing the template

Art Unit: 2129

being edited as taught by **Habraken** for the benefit of not wanting to lose any recent changes (**Habraken** page 15) since you don't want to lose your valuable documents as you create them (**Habraken** page 13).

Claim 26:

5 The combination of **Chikirivao** and **Wallace** fails to teach:

- Wherein the predefined event is passage of a predetermined amount of time.

Habraken teaches:

- Wherein the predefined event is passage of a predetermined amount of time (pages 4-16 especially "AutoSave feature ... AutoRecoverInfo Every ... set the time interval between autosaves" page 13).

10 Motivation:

Habraken and the combination of **Chikirivao** and **Wallace** are from the same field of endeavor, software. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the combined teachings of **Chikirivao** and **Wallace** by saving information occurs after a predetermined amount of time as taught by **Habraken** for the benefit of not wanting to lose any recent changes (**Habraken** page 15) since you don't want to lose your valuable documents as you create them, so if you are really absent-minded about periodically saving your work, use the AutoSave feature (**Habraken** page 13).

15

Response to Arguments

Applicant has not separately argued the claims in this rejection. The rejection of the claims upon which these claims depend has been maintained, and this rejection is maintained for the same reasons, as detailed above.

20

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

25

Art Unit: 2129

Claims 1, 13-15, 22-35, & 40 are rejected.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to BENJAMIN BUSS whose telephone number is (571)272-5831. The examiner can normally be reached on at least Monday, Tuesday, Thursday, or Friday 9AM-5PM.

As detailed in MPEP 502.03, communications via Internet e-mail are at the discretion of the applicant. Without a written authorization by applicant in place, the USPTO will not respond via Internet e-mail to any Internet correspondence which contains information subject to the confidentiality requirement as set forth in 35 U.S.C. 122. A paper copy of such correspondence will be placed in the appropriate patent application. The following is a sample authorization form which may be used by applicant:

“Recognizing that Internet communications are not secure, I hereby authorize the USPTO to communicate with me concerning any subject matter of this application by electronic mail. I understand that a copy of these communications will be made of record in the application file.”

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, David Vincent can be reached on 571-272-3080. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Benjamin Buss
Examiner
Art Unit 2129

/B. B./
Examiner, Art Unit 2129
/David R Vincent/
Supervisory Patent Examiner, Art Unit 2129